

# A Generalized Reduced Linear Program for Markov Decision Processes

Chandrashekar Lakshminarayanan and Shalabh Bhatnagar,  
 Department Computer Science and Automation,  
 Indian Institute of Science, Bangalore-560012, India.  
 {chandrul,shalabh}@csa.iisc.ernet.in

November 19, 2014

## Abstract

Markov decision processes (MDPs) with large number of states are of high practical interest. However, conventional algorithms to solve MDP are computationally infeasible in this scenario. Approximate dynamic programming (ADP) methods tackle this issue by computing approximate solutions. A widely applied ADP method is approximate linear program (ALP) which makes use of linear function approximation and offers theoretical performance guarantees. Nevertheless, the ALP is difficult to solve due to the presence of a large number of constraints and in practice, a reduced linear program (RLP) is solved instead. The RLP has a tractable number of constraints sampled from the original constraints of the ALP. Though the RLP is known to perform well in experiments the theoretical guarantees are available only for a specific RLP obtained under idealized assumptions. In this paper, we generalize the RLP to define a generalized reduced linear program (GRLP) which has a tractable number of constraints that are obtained as positive linear combinations of the original constraints of the ALP. The main contribution of this paper is the novel theoretical framework developed to obtain error bounds for any given GRLP. Central to our framework are two max-norm contraction operators. Our result solves theoretically justifies linear approximation of constraints. We discuss the implication of our results in the contexts of ADP and reinforcement learning. We also demonstrate via an example in the domain of controlled queues that the experiments conform to the theory.

## 1 Introduction

Markov decision processes (MDPs) is an important mathematical framework to study optimal sequential decision making problems that arise in science and engineering. Solving an MDP involves computing the optimal *value-function* ( $J^*$ ), a vector whose dimension is the number of states. MDPs with small number of states can be solved easily by conventional solution methods such as value/ policy iteration or linear programming (LP) [2]. Dynamic programming is at the heart of all the conventional solution methods for MDPs.

The term *curse-of-dimensionality* (or in short *curse*) denotes the fact that the number of states grows exponentially in the number of state variables. Most practical MDPs suffer from the curse, i.e., have large number of states and the  $J^*$  is difficult to compute. A practical way to tackle the curse is to compute an approximate value function  $\tilde{J}$  instead of  $J^*$ . The methods that compute  $\tilde{J}$  instead of  $J^*$  are known as approximate dynamic programming (ADP) methods whose success depends on the quality of approximation, i.e., on the quantity  $\|J^* - \tilde{J}\|$ . Most ADP methods employ linear function approximation (LFA), i.e., let  $\tilde{J} = \Phi r^*$ , where  $\Phi$  is a feature matrix and  $r^*$  is a learnt weight vector. Dimensionality reduction is achieved by choosing  $\Phi$  to have fewer columns in comparison to the number of states and this makes computing  $\tilde{J}$  easier.

Approximate linear program (ALP) [6, 7, 8, 4, 11, 22, 18] employs LFA in the linear programming formulation ([2, 1]) of MDP. The ALP computes an approximate value function and offers sound theoretical guarantees. A serious shortcoming of the ALP is the large number of constraints (of the order of the number of states). A technique studied in literature that tackles the issue of large number of constraints is constraint sampling [7, 10] wherein one solves a reduced linear program (RLP) with a small number of constraints sampled from the constraints of the ALP. [7] presents performance guarantees for the RLP when the constraints are sampled with respect to the stationary distribution of the optimal policy. Such an idealized assumption on the availability of the optimal policy (which in turn requires knowledge of  $J^*$ ) is a shortcoming. Nevertheless, the RLP has been shown to perform empirically well ([7, 6, 8]) even when the constraints are not sampled using the stationary distribution of the optimal policy. Motivated by the gap between the limited theoretical guarantees of the RLP and its successful practical efficacy, in this paper we provide a novel theoretical framework to characterize the error due to constraint reduction/approximation. The novelty and salient points of our contributions are listed below:

1. We define a generalized reduced linear program (GRLP) which has a tractable number of constraints that are obtained as positive linear combinations of the original constraints of the ALP.
2. We develop a novel analytical framework in order to relate  $\hat{J}$ , the solution to the GRLP, and the optimal value function  $J^*$ . In particular, we come up with two novel max-norm contraction operators called the least upper bound (LUB) projection operator and the approximate least upper bound projection operator (ALUB).
3. We show that  $\|J^* - \hat{J}\| \leq (c_1 + c_2)$ , where  $c_1 > 0$ ,  $c_2 > 0$  are constants. While the term  $c_1$  corresponds to the error inherent to the ALP itself, the term  $c_2$  constitutes the additional error introduced due to constraint approximation.
4. The results from the GRLP framework solves the problem of theoretically justifying linear approximation of constraints. Unlike the bounds in [7] that hold only for specific RLP our bounds hold for any GRLP and as a result any RLP.
5. We also discuss qualitatively the relative importance of our results in the context of ADP and their implication in the reinforcement learning setting.

6. We demonstrate via an example in controlled queues that the experiments conform to the theory developed.

The rest of the paper is organized as follows. First, we present the basics of MDPs. We then discuss the ALP technique, the basic error bounds as well as, the issues and proposed solutions in literature, following by which we present the open questions we solve in this paper. We then present the main results of the paper namely the GRLP and its error analysis. We then present a qualitative discussion of our result followed by the numerical example.

## 2 Markov Decision Processes (MDPs)

In this section, we briefly discuss the basics of Markov Decision Processes (MDPs) (the reader is referred to [2, 20] for a detailed treatment).

**The MDP Model:** An MDP is a 4-tuple  $\langle S, A, P, g \rangle$ , where  $S$  is the state space,  $A$  is the action space,  $P$  is the probability transition kernel and  $g$  is the reward function. We consider MDPs with large but finite number of states, i.e.,  $S = \{1, 2, \dots, n\}$  for some large  $n$ , and the action set is given by  $A = \{1, 2, \dots, d\}$ . For simplicity, we assume that all actions are feasible in all states. The probability transition kernel  $P$  specifies the probability  $p_a(s, s')$  of transitioning from state  $s$  to state  $s'$  under the action  $a$ . We denote the reward obtained for performing action  $a \in A$  in state  $s \in S$  by  $g_a(s)$ .

**Policy:** A policy  $\mu$  specifies the action selection mechanism, and is described by the sequence  $\mu = \{u_1, u_2, \dots, u_n, \dots\}$ , where  $u_n: S \rightarrow A, \forall n \geq 0$ . A stationary deterministic policy (SDP) is one where  $u_n \equiv u, \forall n \geq 0$  for some  $u: S \rightarrow A$ . By abuse of notation we denote the SDP by  $u$  itself instead of  $\mu$ . In the setting that we consider, one can find an SDP that is optimal [2, 20]. In this paper, we restrict our focus to the class  $U$  of SDPs. Under an SDP  $u$ , the MDP is a Markov chain with probability transition kernel  $P_u$ .

**Value Function:** Given an SDP  $u$ , the infinite horizon discounted reward corresponding to state  $s$  under  $u$  is denoted by  $J_u(s)$  and is defined by

$$J_u(s) \triangleq \mathbf{E}\left[\sum_{n=0}^{\infty} \alpha^n g_{a_n}(s_n) \mid s_0 = s, a_n = u(s_n) \forall n \geq 0\right],$$

where  $\alpha \in (0, 1)$  is a given discount factor. Here  $J_u(s)$  is known as the value of the state  $s$  under the SDP  $u$ , and the vector quantity  $J_u \triangleq (J_u(s), \forall s \in S) \in R^n$  is called the value-function corresponding to the SDP  $u$ .

**The optimal policy**  $u^*$  is obtained as  $u^*(s) \triangleq \arg \max_{u \in U} J_u(s)$ <sup>1</sup>.

**The optimal value-function**  $J^*$  is the one obtained under the optimal policy, i.e.,  $J^* = J_{u^*}$ .

**The Bellman Equation and Operator:** Given an MDP, our aim is to find the optimal

---

<sup>1</sup>Such  $u^*$  exists and is well defined in the case of infinite horizon discounted reward MDP, for more details see [20].

value function  $J^*$  and the optimal policy  $u^*$ . The optimal policy and value function obey the Bellman equation (BE) as under:  $\forall s \in S$ ,

$$J^*(s) = \max_{a \in A} (g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s')), \quad (1a)$$

$$u^*(s) = \arg \max_{a \in A} (g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s')). \quad (1b)$$

Typically  $J^*$  is computed first and  $u^*$  is obtained by substituting  $J^*$  in (1b). The Bellman operator  $T: \mathbf{R}^n \rightarrow \mathbf{R}^n$  is defined using the model parameters of the MDP as follows:

$$(TJ)(s) = \max_{a \in A} (g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s')), \text{ where } J \in \mathbf{R}^n.$$

**Basis Solution Methods:** When the number of states of the MDP is small,  $J^*$  and  $u^*$  can be computed exactly using conventional methods such as value/policy iteration and linear programming (LP) [2].

**Curse-of-Dimensionality** is a term used to denote the fact that the number of states grows exponentially in the number of state variables. Most MDPs occurring in practice suffer from the curse, i.e., have large number of states and it is difficult to compute  $J^* \in \mathbf{R}^n$  exactly in such scenarios.

**Approximate Dynamic Programming** [13, 17, 6, 24](ADP) methods compute an approximate value function  $\tilde{J}$  instead of  $J^*$ . In order to make the computations easier ADP methods employ function approximation (FA) where  $\tilde{J}$  is chosen from a parameterized family of functions. The problem then boils down to finding the optimal parameter which is usually of lower dimension and is easily computable.

**Linear Function Approximation (LFA)** [6, 17, 12, 14, 15] is a widely used FA scheme such that the approximate value function  $\tilde{J} = \Phi r^*$ , where  $\Phi = [\phi_1 | \dots | \phi_k]$  is an  $n \times k$  feature matrix and  $r^*$  is the parameter to be learnt.

### 3 Approximate Linear Programming

We now present the linear programming formulation of the MDP which forms the basis for ALP. The LP formulation is obtained by unfurling the max operator in the BE in (1) into a set of linear inequalities as follows:

$$\begin{aligned} \min_{J \in \mathbf{R}^n} & c^\top J \\ \text{s.t } & J(s) \geq g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s'), \forall s \in S, a \in A, \end{aligned} \quad (2)$$

where  $c \in \mathbf{R}_+^n$  is a probability distribution and denotes the relative importance of the various states. One can show that  $J^*$  is the solution to (2) [2]. The LP formulation in

(2) can be represented in short<sup>2</sup> as,

$$\begin{aligned} \min_{J \in \mathbf{R}^n} c^\top J \\ \text{s.t } J \geq TJ. \end{aligned} \quad (3)$$

The approximate linear program (ALP) is obtained by making use of LFA in the LP, i.e., by letting  $J = \Phi r$  in (3) and is given as

$$\begin{aligned} \min_{r \in \mathbf{R}^k} c^\top \Phi r \\ \text{s.t } \Phi r \geq T\Phi r. \end{aligned} \quad (4)$$

Unless specified otherwise we use  $\tilde{r}_c$  to denote the solution to the ALP and  $\tilde{J}_c = \Phi \tilde{r}_c$  to denote the corresponding approximate value function. The following is a preliminary error bound for the ALP from [6]:

**Theorem 1** *Let 1, i.e., the vector with all-components equal to 1, be in the span of the columns of  $\Phi$  and  $c$  be a probability distribution. Then, if  $\tilde{J}_c = \Phi \tilde{r}_c$  is an optimal solution to the ALP in (4), then  $\|J^* - \tilde{J}_c\|_{1,c} \leq \frac{2}{1-\alpha} \min_r \|J^* - \Phi r\|_\infty$ , where  $\|x\|_{1,c} = \sum_{i=1}^n c(i)|x(i)|$ .*

For a more detailed treatment of the ALP and sophisticated bounds the reader is referred to [6]. Note that the ALP is a linear program in  $k$  ( $\ll n$ ) variables as opposed to the LP in (3) which has  $n$  variables. Nevertheless, the ALP has  $nd$  constraints (same as the LP) which is an issue when  $n$  is large and calls for constraint approximation/reduction techniques.

### 3.1 Related Work

**Constraint sampling and The RLP:** The most important work in the direction of constraint reduction is constraint sampling [7] wherein a reduced linear program (RLP) is solved instead of the ALP. While the objective of the RLP is same as that of the ALP, the RLP has only  $m \ll nd$  constraints. These  $m$  constraints are *sampled* from the original  $nd$  constraints of the ALP according to a special sampling distribution  $\psi_{u^*, V}$ , where  $u^*$  is the optimal policy and  $V$  is a Lyapunov function (see [7] for a detailed presentation). If  $\tilde{r}$  and  $\tilde{r}_{RLP}$  are the solutions to the ALP and the RLP respectively from [7] we know that  $\|J^* - \Phi \tilde{r}_{RLP}\|_{1,c} \leq \|J^* - \Phi \tilde{r}\|_{1,c} + \epsilon \|J^*\|_{1,c}$ . A major gap in the theoretical analysis is that the error bounds are known for only a specific RLP formulated using idealized assumptions, i.e., knowledge of  $u^*$ .

**Other works:** Most works in literature make use of the underlying structure of the problem to cleverly reduce the number of constraints of the ALP. A good example is [11], wherein the structure in factored linear functions is exploited. The use of basis function also helps constraint reduction in [16]. In [4] the constraints are approximated indirectly by approximating the square of the Lagrange multipliers. [19] reduces the

---

<sup>2</sup> $J \geq TJ$  is a shorthand for the  $nd$  constraints in (2). It is also understood that constraints  $(i-1)n + 1, \dots, in$  correspond to the  $i^{th}$  action.

transitional error ignoring the representational and sampling errors. Empirical successes include repeated application of constraint sampling to solve Tetris [10].

**Open Questions:** The fact that RLP works well empirically goads us to build a more elaborate theory for constraint reduction. In particular, one would like to answer the following questions related to constraint reduction in ALP that have so far remained open.

- As a natural generalization of the RLP, what happens if we define a generalized reduced linear program (GRLP) whose constraints are positive linear combinations of the original constraints of the ALP?
- Unlike [7] which provides error bounds for a specific RLP formulated using an idealized sampling distribution is it possible to provide error bounds for any GRLP (and as result any RLP)? In this paper, we address both of the questions above.

## 4 Generalized Reduced Linear Program

We define the generalized reduced linear program (GRLP) as below:

$$\begin{aligned} \min_{r \in \chi} & c^\top \Phi r, \\ \text{s.t. } & W^\top \Phi r \geq W^\top T \Phi r, \end{aligned} \quad (5)$$

where  $W \in \mathbf{R}_+^{nd \times m}$  is an  $nd \times m$  matrix with all positive entries and  $\chi \subset \mathbf{R}^k$  is any bounded set such that  $\hat{J}_c \in \chi$ . Thus the  $i^{th}$  ( $1 \leq i \leq m$ ) constraint of the GRLP is a positive linear combination of the original constraints of the ALP, see Assumption 1. Constraint reduction is achieved by choosing  $m \ll nd$ . Unless specified otherwise we use  $\hat{r}_c$  to denote the solution to the GRLP in (5) and  $\hat{J}_c = \Phi \hat{r}_c$  to denote the corresponding approximate value function. We assume the following throughout the rest of the paper:

**Assumption 1**  $W \in \mathbf{R}_+^{nd \times m}$  is a full rank  $nd \times m$  matrix with all non-negative entries. The first column of the feature matrix  $\Phi$  (i.e.,  $\phi_1$ ) is  $\mathbf{1}^3 \in \mathbf{R}^n$  and that  $c = (c(i), i = 1, \dots, n) \in \mathbf{R}^n$  is a probability distribution, i.e.,  $c(i) \geq 0$  and  $\sum_{i=1}^n c(i) = 1$ . It is straightforward to see that a RLP is trivially a GRLP.

As a result of constraint reduction the feasible region of the GRLP is a superset of the feasible region of the ALP (see Figure 1). In order to bound  $\|J^* - \hat{J}_c\|$ , [6] makes use of the property that  $\Phi \tilde{r}_c \geq T \Phi \tilde{r}_c$ . However in the case of the GRLP this property does not hold anymore and hence it is a challenge to bound the error  $\|J^* - \hat{J}_c\|$ . We tackle this challenge by introducing two novel max-norm contraction operators called the least upper bound projection (LUBP) and approximate least upper bound projection operators (ALUBP) denoted by  $\Gamma$  and  $\tilde{\Gamma}$  respectively. We first present some definitions before the main result and a sketch of its proof. The least upper bound (LUB) projection operator  $\Gamma: \mathbf{R}^n \rightarrow \mathbf{R}^n$  is defined as below:

---

<sup>3</sup> $\mathbf{1}$  is a vector with all components equal to 1. This definition is used throughout the paper.

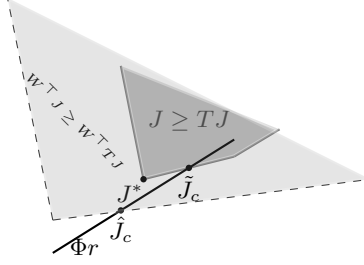


Figure 1: The outer lightly shaded region corresponds to GRLP constraints and the inner dark shaded region corresponds to the original constraints. The main contribution of the paper is to provide a bound for  $\|J^* - \hat{J}_c\|$ .

**Definition 2** Given  $J \in \mathbf{R}^n$ , its least upper bound projection is denoted by  $\Gamma J$  and is defined as

$$(\Gamma J)(i) \triangleq \min_{j=1, \dots, k} (\Phi r_{e_j})(i), \quad \forall i = 1, \dots, n, \quad (6)$$

where  $V(i)$  denotes the  $i^{\text{th}}$  component of the vector  $V \in \mathbf{R}^n$ . Also in (6),  $e_j$  is the vector with 1 in the  $j^{\text{th}}$  place and zeros elsewhere, and  $r_{e_j}$  is the solution to the linear program in (7) for  $c = e_j$ .

$$\begin{aligned} r_c &\triangleq \min_{r \in \mathcal{X}} c^\top \Phi r, \\ &\text{s.t. } \Phi r \geq T J. \end{aligned} \quad (7)$$

**Remark 1**

1. Observe that  $\Gamma J \geq T J$  (follows from the fact that if  $a \geq c$  and  $b \geq c$  then  $\min(a, b) \geq c$ , where  $a, b, c \in \mathbf{R}$ ).
2. Given  $\Phi$  and  $J \in \mathbf{R}^n$ , define  $\mathcal{F} \triangleq \{\Phi r \mid \Phi r \geq T J\}$ . Thus  $\mathcal{F}$  is the set of all vectors in the span of  $\Phi$  that upper bound  $T J$ . By fixing  $c$  in the linear program in (7) we select a unique vector  $\Phi r_c \in \mathcal{F}$ . The LUB projection operator  $\Gamma$  picks  $n$  vectors  $\Phi r_{e_i}, i = 1, \dots, n$  from the set  $\mathcal{F}$  and  $\Gamma J$  is obtained by computing their component-wise minimum.
3. Even though  $\Gamma J$  does not belong to the span of  $\Phi$ ,  $\Gamma J$  in some sense collates the various best upper bounds that can be obtained via the linear program in (7).
4. The LUB operator  $\Gamma$  in (6) bears close similarity to the ALP in (4).

We define an approximate least upper bound (ALUB) projection operator which has a structure similar to the GRLP and is an approximation to the LUB operator.

**Definition 3** Given  $J \in \mathbf{R}^n$ , its approximate least upper bound (ALUB) projection is denoted by  $\tilde{\Gamma}J$  and is defined as

$$(\tilde{\Gamma}J)(i) \triangleq \min_{j=1,\dots,k} (\Phi r_{e_j})(i), \forall i = 1, \dots, n, \quad (8)$$

where  $r_{e_j}$  is the solution to the linear program in (9) for  $c = e_j$ , and  $e_j$  is same as in Definition 2.

$$\begin{aligned} r_c &\triangleq \min_{r \in \chi} c^\top \Phi r, \\ \text{s.t } W^\top \Phi r &\geq W^\top T J, W \in \mathbf{R}_+^{nd \times m}. \end{aligned} \quad (9)$$

**Definition 4** The LUB projection of  $J^*$  is denoted by  $\bar{J} = \Gamma J^*$ , and let  $r^* \triangleq \arg \min_{r \in \mathbf{R}^k} \|J^* - \Phi r^*\|$ .

## 4.1 Main Result

### Theorem 5

$$\|J^* - \hat{J}_c\|_{1,c} \leq \frac{6\|J^* - \Phi r^*\|_\infty + 2\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty}{1 - \alpha}. \quad (10)$$

**Proof:** Here we provide a sketch of the proof. Figure 2 gives an idea of the steps that lead to the result. First, one shows that the operators  $\Gamma$  and  $\tilde{\Gamma}$  have the max-norm contraction property with factor  $\alpha$ . As a result, operators  $\Gamma$  and  $\tilde{\Gamma}$  have fixed points  $\tilde{V} \in \mathbf{R}^n$  and  $\hat{V} \in \mathbf{R}^n$  respectively. This leads to the inequalities  $\tilde{J}_c \geq \tilde{V} \geq J^*$  and  $\hat{J}_c \geq \hat{V}$  (see Figure 2), followed by which one can bound the term  $\|J^* - \hat{V}\|_\infty$  and then go on to show that any solution  $\tilde{r}_c$  to the GRLP is also a solution to the program in (28).

$$\begin{aligned} \min_{r \in \chi} \|\Phi r - \hat{V}\|_{1,c} \\ \text{s.t } W^\top \Phi r &\geq W^\top T \Phi r. \end{aligned} \quad (11)$$

One then obtains the bound  $\|J^* - \hat{J}_c\|_{1,c}$  as in (33) using the fact that  $\|J^* - \bar{J}\|_\infty \leq 2\|J^* - \Phi r^*\|_\infty$  where  $r^*$  is as in Definition 4.

It is important to note that computing  $\Gamma/\tilde{\Gamma}$  involves solving  $n$  linear programs which is easy when  $n$  is small, however, the same becomes difficult and impractical when  $n$  is large. Nevertheless, we hasten to point out that these quantities are only analytical constructs that lead us to the error bounds, and need not be calculated in practice for systems with large  $n$ .

## 4.2 Result Discussion

We now make various important qualitative observations about the result in Theorem 5.

**Error Terms:** The error term is split into two factors, the first of which is related to the



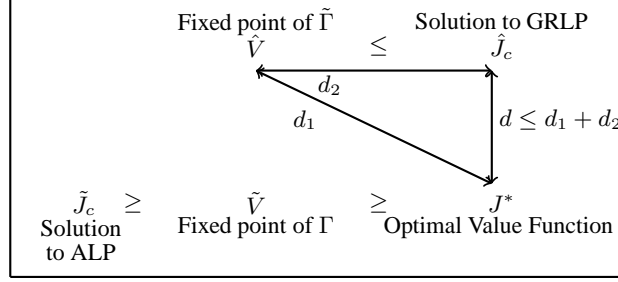


Figure 2: A schematic of the error analysis.

Here  $d = \|\hat{J}_c - J^*\|_{1,c}$ .

best possible projection while the second factor is related to constraint approximation. The second factor  $\|\Gamma\bar{J} - \tilde{\Gamma}\bar{J}\|_\infty$  is completely defined in terms of  $\Phi$ ,  $W$  and  $T$ , and does not require knowledge of stationary distribution of the optimal policy. It makes intuitive sense since given that  $\Phi$  approximates  $J^*$ , it is enough for  $W$  to depend on  $\Phi$  and  $T$  without any additional requirements. Unlike the result in [7] which holds only for a specific RLP formulated under ideal assumptions, our bounds hold for any GRLP and as a result for any given RLP. Another interesting feature of our result is that it holds with probability 1. Also by making use of appropriate Lyapunov functions as in [6], the error bound in (33) can also be stated using a weighted  $L_\infty$ -norm, thereby indicating the relative importance of states.

**Additional insights on constraint sampling:** It is easy to notice from Definitions 2, 3 and 4 that for any given state  $s \in S$ ,  $\Gamma\bar{J}(s) \geq J^*(s)$ , and that  $\Gamma\bar{J}(s) \geq \tilde{\Gamma}\bar{J}(s)$ . If the state  $s$  is selected in the RLP, then it is also true that  $\Gamma\bar{J}(s) \geq \tilde{\Gamma}\bar{J}(s) \geq J^*(s)$ . Thus the additional error  $|\Gamma\bar{J}(s) - \tilde{\Gamma}\bar{J}(s)|$  due to constraint sampling is less than the original projection error  $|\Gamma\bar{J}(s) - J^*(s)|$  due to function approximation. This means that the RLP is expected to perform well whenever *important* states are retained after constraint sampling. Thus the sampling distribution need not be the stationary distribution of the optimal policy as long as it samples the important states, an observation that might theoretically explain the empirical successes of the RLP [6, 10, 8].

**Relation to other ADP methods:**

ADP Method	Empirical	Theoretical
Projected Bellman Equation	✓ [5, 17, 13]	✗-Policy Chattering [2]
ALP	✗-Large Constraints	✓-[6]
RLP	✓-[10, 6, 8]	✗- Only under ideal assumptions

A host of the ADP methods such as [13, 17, 5, 23] are based on solving the projected Bellman equation (PBE). The PBE based methods have been empirically successful and also have theoretical guarantees for the approximate value function. However, a significant shortcoming is that they suffer from issue of *policy-chattering* (see

section 6.4.3 of [2]), i.e., the sequence of policies might oscillate within a set of bad policies. A salient feature of the ALP based methods is that they find only one approximate value function  $\tilde{J}_c$  and one sub-optimal policy derived as a greedy policy with respect to  $\tilde{J}_c$ . As a result there is no such issue of policy-chattering for the ALP based methods. By providing the error bounds for the GRLP, our paper provides the much required theoretical support for the RLP. Our GRLP framework closes the long-standing gap in the literature of providing a theoretical framework to bound the error due to constraint reduction in ALP based schemes.

**GRLP is linear function approximation of the constraints:** In order to appreciate this fact consider the Lagrangian of the ALP and GRLP in (12) and (13) respectively, i.e.,

$$\tilde{L}(r, \lambda) = c^\top \Phi r + \lambda^\top (T\Phi r - \Phi r), \quad (12)$$

$$\hat{L}(r, q) = c^\top \Phi r + q^\top W^\top (T\Phi r - \Phi r). \quad (13)$$

The insight that the GRLP is linear function approximation of constraints (i.e., the Lagrangian multipliers) can be obtained by noting that  $Wq \approx \lambda$  in (13). Note that while the ALP employs LFA in its objective, the GRLP employs linear approximation both in the objective as well as the constraints. This has significance in the context of the reinforcement learning setting [21] wherein the model information is available in the form of noisy sample trajectories. RL algorithms make use of stochastic approximation (SA) [3] and build on ADP methods to come up with incremental update schemes to learn from noisy samples presented to them. An SA scheme to solve the GRLP in RL setting can be derived in a manner similar to [4].

## 5 Application to Controlled Queues

We take up an example in the domain of controlled queues to show that experiments confirm with the theory developed. More specifically, we look at the error bounds for different constraints reduction schemes to demonstrate the fact that whenever value of  $\|\Gamma\tilde{J} - \tilde{\Gamma}\tilde{J}\|_\infty$  is less the GRLP solution is closer the optimal value function.

The queuing system consists of  $n = 10^4$  states and  $d = 4$  actions. We chose  $n = 10^4$  because it was possible to solve both the GRLP and the exact LP (albeit with significant effort) so as to enumerate the approximation errors. We hasten to mention that while we could run the GRLP for queuing systems with  $n > 10^4$  without much computational overhead, solving the exact LP was not possible for  $n > 10^4$  as a result of which the approximation error could not be computed.

**Queuing Model:** The queuing model used here is similar to the one in Section 5.2 of [6]. We consider a single queue with arrivals and departures. The state of the system is the queue length with the state space given by  $S = \{0, \dots, n-1\}$ , where  $n-1$  is the buffer size of the queue. The action set  $A = \{1, \dots, d\}$  is related to the service rates. We let  $s_t$  denote the state at time  $t$ . The state at time  $t+1$  when action  $a_t \in A$  is chosen is given by  $s_{t+1} = s_t + 1$  with probability  $p$ ,  $s_{t+1} = s_t - 1$  with probability  $q(a_t)$  and  $s_{t+1} = s_t$ , with probability  $(1 - p - q(a_t))$ . For states  $s_t = 0$  and  $s_t = n-1$ , the system dynamics is given by  $s_{t+1} = s_t + 1$  with probability  $p$  when

$s_t = 0$  and  $s_{t+1} = s_t - 1$  with probability  $q(a_t)$  when  $s_t = n - 1$ . The service rates satisfy  $0 < q(1) \leq \dots \leq q(d) < 1$  with  $q(d) > p$  so as to ensure ‘stabilizability’ of the queue. The reward associated with the action  $a \in A$  in state  $s \in S$  is given by  $g_a(s) = -(s + 60q(a)^3)$ .

**Choice of  $\Phi$  :** We make use of polynomial features in  $\Phi$  (i.e.,  $1, s, \dots, s^{k-1}$ ) since they are known to work well for this domain [6]. This takes care of the term  $\|J^* - \Phi r^*\|_\infty$  in (33).

**Selection of  $W$  :** For our experiments, we choose two contenders for the  $W$ -matrix:

(i)  $W_c$ - matrix that corresponds to sampling according to  $c$ . This is justified by the insights obtained from the error term  $\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$  and the idea of selecting the important states.

(ii)  $W_a$  state-aggregation matrix, a heuristic derived by interpreting  $W$  to be the feature matrix that approximates the Lagrange multipliers as  $\lambda \approx Wq$ , where  $\lambda \in \mathbf{R}^{nd}$ ,  $r \in \mathbf{R}^m$ . One can show [9] that the optimal Lagrange multipliers are the discounted number of visits to the “state-action pairs” under the optimal policy  $u^*$ , i.e.,

$$\begin{aligned} \lambda^*(s, u^*(s)) &= (c^\top (I - \alpha P_{u^*})^{-1})(s) \\ &= (c^\top (I + \alpha P_{u^*} + \alpha^2 P_{u^*}^2 + \dots))(s). \\ \lambda^*(s, u^*(s)) &= 0, \forall a \neq u^*(s). \end{aligned}$$

where  $P_{u^*}$  is the probability transition matrix with respect to the optimal policy. Even though we might not have the optimal policy in practice  $u^*$ , the fact that  $\lambda^*$  is a linear combination of  $\{P_{u^*}, P_{u^*}^2, \dots\}$  hints at the kind of features that might be useful for the  $W$  matrix. Our choice of  $W_a$  matrix to correspond to aggregation of near by states is motivated by the observation that  $P^n$  captures  $n^{th}$  hop connectivity/neighborhood information. The aggregation matrix  $W_a$  is defined as below:  $\forall i = 1, \dots, m$ ,

$$\begin{aligned} W_a(i, j) &= 1, \forall j \text{ s.t } j = (i - 1) \times \frac{n}{m} + k + (l - 1) \times n, \\ &\quad k = 1, \dots, \frac{n}{m}, l = 1, \dots, d, \\ &= 0, \text{ otherwise.} \end{aligned} \tag{14}$$

In order to provide a contrast between good and bad choices of  $W$  matrices we also make use of two more matrices, an ideal matrix  $W_i$  generated by sampling according to the stationary distribution of the optimal policy as in [7] and  $W_c$  generated by sampling using  $c$  and  $W_r$  a random matrix in  $\mathbf{R}_+^{nd \times m}$ . For the sake of comparison we compute  $\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$  for the different  $W$  matrices. Though computing  $\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$  might be hard in the case of large  $n$ , since  $\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$  is completely dependent on the structure of  $\Phi, T$  and  $W$  we can compute it for small  $n$  instead and use it as a surrogate. Accordingly, we first chose a smaller system  $Q_S$  with  $n = 10, d = 2, k = 2, m = 5, q(1) = 0.2, q(2) = 0.4, p = 0.2$  and  $\alpha = 0.98$ . In the case of  $Q_S, W_a$  ((14) with  $m = 5$ ) turns out to be a  $20 \times 5$  matrix where the  $i^{th}$  constraint of the GRLP is the average of all constraints corresponding to states  $(2i - 1)$  and  $2i$  (there are four constraints corresponding to these two states). The various error terms are listed in Table 1 and plots are shown in Figure 3. It is clear from Table 1 that  $W_a, W_i$  and  $W_c$  have much better  $\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$  than randomly generated positive matrices. Since

each constraint is a hyperplane, taking linear combinations of non-adjacent hyperplanes might drastically affect the final solution. This could be a reason why  $W_r$  (random matrix) performs badly in comparison with other  $W$  matrices.

Error Term	$W_i$	$W_c$	$W_a$	$W_r$
$\ \Gamma\bar{J} - \tilde{\Gamma}\bar{J}\ _\infty$	39	84	54.15	251.83

Table 1: Shows various error terms for  $Q_S$ .

Having validated the choices of  $W$ s on  $Q_S$  we then consider a moderately larger queuing system (denoted by)  $Q_L$  with  $n = 10^4$  and  $d = 4$  with  $q(1) = 0.2$ ,  $q(2) = 0.4$ ,  $q(3) = 0.6$ ,  $q(4) = 0.8$ ,  $p = 0.4$  and  $\alpha = 0.98$ . In the case of  $Q_L$  we chose  $k = 4$  (i.e., we used  $1, s, s^2$  and  $s^3$  as basis vectors) and we chose  $W_a$  (14),  $W_c$ ,  $W_i$  and  $W_r$  with  $m = 50$ . We set  $c(s) = (1 - \zeta)\zeta^s$ ,  $\forall s = 1, \dots, 9999$ , with  $\zeta = 0.9$  and  $\zeta = 0.999$  respectively. The results in Table 2 show that performance exhibited by  $W_a$  and  $W_c$  are better by several orders of magnitude over ‘random’ in the case of the large system  $Q_L$  and is closer to the ideal sampler  $W_i$ . Also note that a better performance of  $W_a$  and  $W_c$  in the larger system  $Q_L$  tallies with a lower value of  $\|\Gamma\bar{J} - \tilde{\Gamma}\bar{J}\|_\infty$  in the smaller system  $Q_S$ .

Error Terms	$W_i$	$W_c$	$W_a$	$W_r$
$\ J^* - \hat{J}_c\ _{1,c}$ for $\zeta = 0.9$	32	32	220	$5.04 \times 10^4$
$\ J^* - \hat{J}_c\ _{1,c}$ for $\zeta = 0.999$	110	180.5608	82	$1.25 \times 10^7$

Table 2: Shows performance metrics for  $Q_L$ .

## 6 Conclusion

Solving MDPs with large number of states is of practical interest. However, when the number of states is large, it is difficult to calculate the exact value function. ALP is a widely studied ADP scheme that computes an approximate value function and offers theoretical guarantees. Nevertheless, the ALP is difficult to solve due to its large number of constraints and in practice a reduced linear program (RLP) is solved. Though RLP has been shown to perform well empirically, theoretical guarantees are available only for a specific RLP formulated under idealized assumptions. This paper provided a more elaborate treatment of constraint reduction/approximation. Specifically, we generalized the RLP to formulate a generalized reduced linear program (GRLP) and provided error bounds. Our results solved a major open problem of analytically justifying linear function approximation of the constraints. We discussed the implications of our results in the contexts of ADP and reinforcement learning. We demonstrated the fact that experiments conform to the theory developed in this paper via an example in the domain of controlled queues. Future directions include providing more sophisticated error bounds based on Lyapunov functions, a two-time scale actor-critic scheme to solve the GRLP, and basis function adaptation schemes to tune the  $W$  matrix.

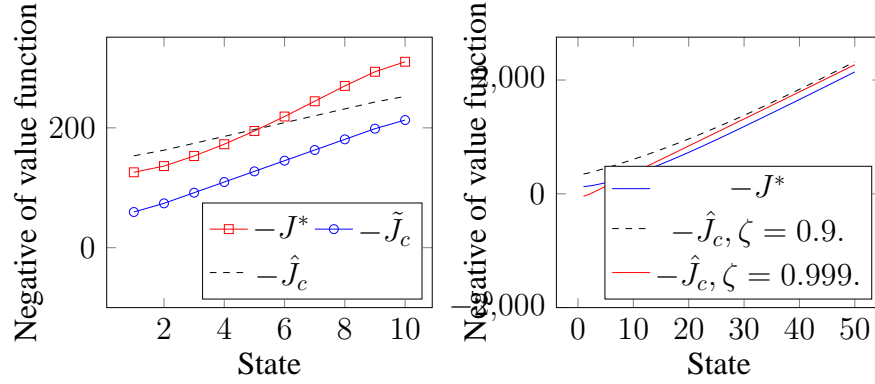


Figure 3: Plot corresponding to  $Q_S$  on the left and  $Q_L$  on the right. The GRLP here used  $W_a$  in (14) with  $m = 5$  for  $Q_S$  and  $m = 50$  for  $Q_L$ .

## References

- [1] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996.
- [2] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, Belmont, MA, 4<sup>th</sup> edition, 2013.
- [3] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. TRIM, 2008.
- [4] V. S. Borkar, J. Pinto, and T. Prabhu. A new learning algorithm for optimal stopping. *Discrete Event Dynamic Systems*, 19(1):91–113, 2009.
- [5] Justin A Boyan. Least-squares temporal difference learning. In *ICML*, pages 49–56. Citeseer, 1999.
- [6] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.
- [7] D. P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Math. Oper. Res.*, 29(3):462–478, 2004.
- [8] V. V. Desai, V. F. Farias, and C. C. Moallemi. A smoothed approximate linear program. In *NIPS*, pages 459–467, 2009.
- [9] D. A. Dolgov and E. H. Durfee. Symmetric approximate linear programming for factored mdps with application to constrained problems. *Annals of Mathematics and Artificial Intelligence*, 47(3-4):273–293, August 2006.

- [10] V. F. Farias and B. Van Roy. Tetris: A study of randomized constraint sampling. In *Probabilistic and Randomized Methods for Design Under Uncertainty*, pages 189–201. Springer, 2006.
- [11] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *J. Artif. Intell. Res.(JAIR)*, 19:399–468, 2003.
- [12] G. Konidaris, S. Osentoski, and P. S. Thomas. Value function approximation in reinforcement learning using the Fourier basis. In *AAAI*, 2011.
- [13] M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.
- [14] S. Mahadevan and B. Liu. Basis construction from power series expansions of value functions. In *Advances in Neural Information Processing Systems*, pages 1540–1548, 2010.
- [15] S. S. Mahadevan and M. Maggioni. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision Processes. *Journal of Machine Learning Research*, 8(16):2169–2231, 2007.
- [16] J.R. Morrison and P.R. Kumar. New linear program performance bounds for queueing networks. Technical Report 3, Journal of Optimization Theory and Applications, 1997.
- [17] A. Nedić and D. P. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, 13(1-2):79–110, 2003.
- [18] J. Papis and R. Parr. Non-parametric approximate linear programming for MDPs. In *AAAI*, 2011.
- [19] M. Petrik and S. Zilberstein. Constraint relaxation in approximate linear programs. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 809–816. ACM, 2009.
- [20] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Programming*. John Wiley, New York, 1994.
- [21] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.
- [22] G. Taylor, M. Petrik, R. Parr, and S. Zilberstein. Feature selection using regularization in approximate linear programs for Markov decision processes. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, Haifa, Israel, 2010.
- [23] John N. Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. Technical report, IEEE Transactions on Automatic Control, 1997.

- [24] Y. Wang, B. O'Donoghue, and S. Boyd. Approximate dynamic programming via iterated Bellman inequalities. *International Journal of Robust and Nonlinear Control*, 2014.

## A Proofs

We present the proofs for the Lemmas and Theorems stated in the main body of the paper. As and when required we also state and prove other intermediate lemmas. For the sake of clarity we restate Assumption 1 as Assumption 2- 4 below:

**Assumption 2**  $W \in \mathbf{R}_+^{nd \times m}$  is a full rank  $nd \times m$  matrix (where  $m \ll nd$ ) with all non-negative entries, and  $\Phi$  is an  $n \times k$  feature matrix (where  $k \ll n$ ).

**Assumption 3** The first column of the feature matrix  $\Phi$  (i.e.,  $\phi_1$ ) is  $\mathbf{1} \in \mathbf{R}^n$ . In other words, the constant function is part of the basis.

**Assumption 4**  $c = (c(i), i = 1, \dots, n) \in \mathbf{R}^n$  is a probability distribution, i.e.,  $c(i) \geq 0$  and  $\sum_{i=1}^n c(i) = 1$ .

We now state without proof the following properties of  $T$ :

**Lemma 6 Monotonicity:** Let  $J_1, J_2 \in \mathbf{R}^n$  be such that  $J_1 \geq J_2$ , then  $TJ_1 \geq TJ_2$ .

**Lemma 7 Shifting:** For any  $J \in \mathbf{R}^n$  and  $\mathbf{1} \in \mathbf{R}^n$  be a vector with all components 1<sup>4</sup> and  $k \in \mathbf{R}$  be a constant, then  $T(J + k\mathbf{1}) = TJ + \alpha k\mathbf{1}$ .

**Lemma 8 Contraction:** For any  $J_1, J_2 \in \mathbf{R}^n$ ,  $\|TJ_1 - TJ_2\|_\infty \leq \alpha \|J_1 - J_2\|_\infty$ .

Lemmas 6, 7 and 8 are standard in MDP literature and can be found in [2].

**Lemma 9** The RLP in (2) of [7] obtained via sampling the constraints is a special case of GRLP.

**Proof:** Let the constraints of the ALP be numbered from  $1 \rightarrow nd$  and  $q_1, \dots, q_m$  denote the  $m$  sampled constraints. The GRLP with  $W$  defined as

$$\begin{aligned} W(i, j) &= 1, \text{ if } q_i = j \\ &= 0, \text{ otherwise} \end{aligned} \tag{15}$$

is the RLP with the corresponding sampled constraints.

**Lemma 10** Let  $r_f \in \mathbf{R}^k$  be any feasible solution to the ALP in (4), then it is also feasible for the GRLP in (5).

**Proof:** Follows from the fact that  $W$  has all positive entries and that each constraint of the GRLP is a positive linear combination of original constraints in the ALP.

**Lemma 11** Let  $r^* \in \mathbf{R}^k$  be defined as  $r^* \triangleq \arg \min_{r \in \mathbf{R}^k} \|J^* - \Phi r\|_\infty$ , then

$$\|J^* - \bar{J}\|_\infty \leq 2\|J^* - \Phi r^*\|_\infty. \tag{16}$$

---

<sup>4</sup>This definition of  $\mathbf{1}$  is the same throughout the paper.



**Proof:** The result follows from the definition of  $\Gamma$  in (6), Assumption 3 and the fact that  $\Phi r^* + \|J^* - \Phi r^*\|_\infty \mathbf{1} \geq TJ^*$ .

**Lemma 12** For  $J_1, J_2 \in \mathbf{R}^n$  such that  $J_1 \geq J_2$ , we have  $\Gamma J_1 \geq \Gamma J_2$ .

**Proof:** Choose any  $i \in \{1, \dots, n\}$  and let  $r_{e_i}^1$  and  $r_{e_i}^2$  be the unique solutions to the linear program in (7) for  $c = e_i$  with  $J = J_1$  and  $J = J_2$  respectively. Since  $J_1 \geq J_2$ , we have  $TJ_1 \geq TJ_2$  and  $e_i^\top \Phi r_{e_i}^1 \geq e_i^\top \Phi r_{e_i}^2$ , i.e.,  $(\Phi r_{e_i}^1)(i) \geq (\Phi r_{e_i}^2)(i)$ . The proof follows from the fact that  $(\Gamma J)(i) = (\Phi r_{e_i})(i)$ ,  $\forall J \in \mathbf{R}^n$ , and our choice of  $i$  was arbitrary.

**Lemma 13** Let  $J_1 \in \mathbf{R}^n$  and  $k \in \mathbf{R}$  be a constant. If  $J_2 = J_1 + k\mathbf{1}$ , then  $\Gamma J_2 = \Gamma J_1 + \alpha k\mathbf{1}$ .

**Proof:** Choose any  $i \in \{1, \dots, n\}$ , let  $r_{e_i}^1$  and  $r_{e_i}^2$  be the unique solutions to linear program in (7) for  $c = e_i$  with  $J = J_1$  and  $J = J_2$  respectively. By Assumption 3 and Lemma 7, we know that  $r_{e_i}^1 + \alpha k e_1$  is feasible for the  $i^{th}$  linear program associated with  $\Gamma J_2$  and we claim that  $r_{e_i}^2 = r_{e_i}^1 + \alpha k e_1$ . On the contrary, if  $r_{e_i}^2 \neq r_{e_i}^1 + \alpha k e_1$ , then  $(\Phi r_{e_i}^2)(i) < (\Phi r_{e_i}^1 + \alpha k e_1)(i)$  (since the solution to the linear program in (7) is unique) and since  $r_{e_i}^2 - \alpha k e_1$  is feasible for the  $i^{th}$  linear program associated with  $\Gamma J_1$  we will have  $(\Phi r_{e_i}^2 - \alpha k e_1)(i) < (\Phi r_{e_i}^1)(i)$ . Thus we have arrived at a contradiction because we assumed that  $r_{e_i}^1$  is the unique solution for the  $i^{th}$  linear program associated with  $\Gamma J_1$ . So

$$r_{e_i}^2 = r_{e_i}^1 + \alpha k e_1, \forall i \in \{1, \dots, n\}, \text{ since } i \text{ was arbitrary.} \quad (17)$$

From (17) and Assumption 3 it follows that  $\Gamma J_2 = \Gamma J_1 + \alpha k\mathbf{1}$ .

**Theorem 14** The operator  $\Gamma: \mathbf{R}^n \rightarrow \mathbf{R}^n$  obeys the max-norm contraction property with factor  $\alpha$ .

**Proof:** Given  $J_1, J_2 \in \mathbf{R}^n$  let  $\epsilon = \|J_1 - J_2\|_\infty$ . Thus

$$J_2 - \epsilon \mathbf{1} \leq J_1 \leq J_2 + \epsilon \mathbf{1}. \quad (18)$$

From Lemmas 12 and 13 we can write

$$\Gamma J_2 - \alpha \epsilon \mathbf{1} \leq \Gamma J_1 \leq \Gamma J_2 + \alpha \epsilon \mathbf{1}. \quad (19)$$

One can show that the following iterative scheme in (20) based on the LUB projection operator  $\Gamma$  in (6) converges to a unique fixed point  $\tilde{V}$ .

$$V_{n+1} = \Gamma V_n, \forall n \geq 0. \quad (20)$$

**Lemma 15**  $\tilde{V}$ , the unique fixed point of the iterative scheme (20), obeys  $\tilde{V} \geq T\tilde{V}$ .

**Proof:** Consider the  $i^{th}$  linear program associated with  $\Gamma\tilde{V}$ . We know that  $\Phi r_{e_i} \geq T\tilde{V}$ ,  $\forall i = 1 \rightarrow n$ . The result follows from noting that  $\tilde{V}$  is an unique fixed point of  $\Gamma$  and that  $\tilde{V}(i) = \min_{j=1 \rightarrow n} (\Phi r_{e_j})(i)$ .

**Lemma 16**  $\tilde{V}$ , the unique fixed point of the iterative scheme (20), and the solution  $\tilde{J}_c$  to the ALP in (4), obey the relation  $\tilde{J}_c \geq \tilde{V} \geq J^*$ .

**Proof:** Since  $\tilde{V} \geq T\tilde{V}$  it follows that  $\tilde{V} \geq J^*$ . Let  $\Phi r_1, \Phi r_2, \dots, \Phi r_n$  be solutions to the ALP in (4) for  $c = e_1, e_2, \dots, e_n$  respectively. Now consider the iterative scheme in (20) with  $V_0(i) = \min_{j=1 \rightarrow n} (\Phi r_j)(i)$ . It is clear from the definition of  $V_0$  that  $\tilde{J}_c \geq V_0$ . Also from monotone property of  $T$  we have  $\Phi r_i \geq T\Phi r_i \geq TV_0$ ,  $\forall i = 1 \rightarrow n$  and hence  $V_0 \geq TV_0$ . Since  $V_1 = \Gamma V_0$ , from the definition of  $\Gamma$  in (6) we have  $V_0 \geq V_1$ , and recursively  $V_n \geq V_{n+1}$ ,  $\forall n \geq 0$ . So it follows that  $\tilde{J}_c \geq V_0 \geq V_1 \dots \geq \tilde{V}$ .

**Theorem 17** Let  $\tilde{V}$  be the fixed point of the iterative scheme in (20) and let  $\bar{J}$  be the best possible projection of  $J^*$  as in Definition 4, then

$$\|J^* - \tilde{V}\|_\infty \leq \frac{1}{1 - \alpha} \|J^* - \bar{J}\|_\infty. \quad (21)$$

**Proof:** Let  $\epsilon = \|J^* - \bar{J}\|_\infty$ , and  $\{V_n\}$ ,  $n \geq 0$  be the iterates of the scheme in (20) with  $V_0 = \bar{J}$ , then

$$\begin{aligned} \|J^* - \tilde{V}\|_\infty &\leq \|J^* - V_0 + V_0 - V_1 + V_1 - V_2 + \dots - \tilde{V}\|_\infty \\ &\leq \|J^* - V_0\|_\infty + \|V_0 - V_1\|_\infty + \|V_1 - V_2\|_\infty + \dots \\ &\quad (\text{Since } \|V_1 - V_0\|_\infty = \|\Gamma\bar{J} - \Gamma J^*\|_\infty \leq \alpha \|\bar{J} - J^*\|_\infty, \text{ from Theorem 14}) \\ &\leq \epsilon + \alpha\epsilon + \alpha^2\epsilon + \dots \\ &= \frac{\epsilon}{1 - \alpha}. \end{aligned} \quad (22)$$

**Lemma 18** For  $J_1, J_2 \in \mathbf{R}^n$  such that  $J_1 \geq J_2$ , we have  $\tilde{\Gamma}J_1 \geq \tilde{\Gamma}J_2$ .

**Proof:** Proof follows from Assumptions 2 and 3 using arguments along the lines of Lemma 12.

**Lemma 19** Let  $J_1 \in \mathbf{R}^n$  and  $k \in \mathbf{R}$  be a constant. If  $J_2 = J_1 + k\mathbf{1}$ , then  $\tilde{\Gamma}J_2 = \tilde{\Gamma}J_1 + \alpha k\mathbf{1}$ .

**Proof:** Proof follows from Assumption 2 and 3 using arguments along the lines of Lemma 13.

**Theorem 20** The operator  $\tilde{\Gamma}: \mathbf{R}^n \rightarrow \mathbf{R}^n$  obeys the max-norm contraction property with factor  $\alpha$  and the following iterative scheme based on the ALUB projection operator  $\tilde{\Gamma}$ , see (23), converges to a unique fixed point  $\tilde{V}$ .

$$V_{n+1} = \tilde{\Gamma}V_n, \quad \forall n \geq 0. \quad (23)$$

**Proof:** Follows on similar lines of proof of Theorem 14.

**Lemma 21** *The unique fixed point  $\hat{V}$  of the iteration in (23) and the solution  $\hat{J}_c$  of the GRLP obey  $\hat{J}_c \geq \hat{V}$ .*

**Proof:** Follows in a similar manner as the proof for Lemma 16.

**Theorem 22** *Let  $\hat{V}$  be the fixed point of the iterative scheme in (23) and let  $\bar{J}$  be the best possible approximation of  $J^*$  as in Definition 4, then*

$$\|J^* - \hat{V}\|_\infty \leq \frac{\|J^* - \bar{J}\|_\infty + \|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty}{1 - \alpha}. \quad (24)$$

**Proof:** Let  $\epsilon = \|J^* - \bar{J}\|_\infty$ , and  $\{V_n\}, n \geq 0$  be the iterates of the scheme in (23) with  $V_0 = \bar{J}$ , then

$$\begin{aligned} \|\bar{J} - \tilde{\Gamma} \bar{J}\|_\infty &\leq \|\bar{J} - \Gamma \bar{J}\|_\infty + \|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty \\ &= \|\Gamma J^* - \Gamma \bar{J}\|_\infty + \|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty \\ &\leq \alpha \epsilon + \beta, \end{aligned} \quad (25)$$

where  $\beta = \|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$ . Now

$$\begin{aligned} \|J^* - \hat{V}\|_\infty &\leq \|J^* - V_0 + V_0 - V_1 + V_1 \dots - \hat{V}\|_\infty \\ &\leq \|J^* - V_0\|_\infty + \|V_0 - V_1\|_\infty + \|V_1 - V_2\|_\infty + \dots \\ &= \|J^* - V_0\|_\infty + \|V_0 - V_1\|_\infty + \|\tilde{\Gamma} V_0 - \tilde{\Gamma} V_1\|_\infty + \dots \\ &\leq \epsilon + (\beta + \alpha \epsilon) + \alpha(\beta + \alpha \epsilon) + \dots \\ &= \frac{\epsilon + \beta}{1 - \alpha}. \end{aligned} \quad (26)$$

**Theorem 23** *Let  $\hat{V}$ ,  $\bar{J}$  be as in Theorem 22 and let  $r^* \triangleq \arg \min_{r \in \mathbf{R}^k} \|J^* - \Phi r\|_\infty$  then*

$$\|J^* - \hat{V}\|_\infty \leq \frac{2\|J^* - \Phi r^*\|_\infty + \|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty}{1 - \alpha}. \quad (27)$$

**Proof:** The result is obtained by using Lemma 11 to replace the term  $\|J^* - \bar{J}\|_\infty$  in Theorem 22.

**Lemma 24**  *$\hat{r} \in \mathbf{R}^k$  is a solution to GRLP in (5) iff it solves the following program:*

$$\begin{aligned} \min_{r \in \chi} & \|\Phi r - \hat{V}\|_{1,c} \\ \text{s.t. } & W^\top \Phi r \geq W^\top T \Phi r. \end{aligned} \quad (28)$$

**Proof:** We know from Lemma 21 that  $\hat{J}_c \geq \hat{V}$ , and thus minimizing  $\|\Phi r - \hat{V}\|_{1,c} = \sum_{i=1}^n c(i)|(\Phi r)(i) - \hat{V}(i)| = c^\top \Phi r - c^\top \hat{V}$ , is same as minimizing  $c^\top \Phi r$ .

**Theorem 25** Let  $\hat{V}$  be the solution to the iterative scheme in (23) and let  $\hat{J}_c = \Phi \hat{r}_c$  be the solution to the GRLP. Let  $\bar{J}$  be the best possible approximation to  $J^*$  as in Definition 4, and  $\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$  be the error due to ALUB projection and let  $r^* \triangleq \arg \min \|J^* - \Phi r\|_\infty$ , then

$$\|\hat{J}_c - \hat{V}\|_{1,c} \leq \frac{4\|J^* - \Phi r^*\|_\infty + \|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty}{1 - \alpha}. \quad (29)$$

**Proof:** Let  $\gamma = \|J^* - \Phi r^*\|_\infty$ , then it is easy to see that

$$\begin{aligned} \|J^* - T\Phi r^*\|_\infty &= \|TJ^* - T\Phi r^*\|_\infty \leq \alpha\gamma, \text{ and} \\ \|T\Phi r^* - \Phi r^*\|_\infty &\leq (1 + \alpha)\gamma. \end{aligned} \quad (30)$$

From Assumption 3 there exists  $r' \in \mathbf{R}^k$  such that  $\Phi r' = \Phi r^* + \frac{(1+\alpha)\gamma}{1-\alpha} \mathbf{1}$  and  $r'$  is feasible to the ALP. Now

$$\|\Phi r' - J^*\|_\infty \leq \|\Phi r^* - J^*\|_\infty + \|\Phi r' - \Phi r^*\|_\infty \leq \gamma + \frac{(1+\alpha)\gamma}{1-\alpha} = \frac{2\gamma}{1-\alpha}. \quad (31)$$

Since  $r'$  is also feasible for GRLP in (5) we have

$$\begin{aligned} \|\hat{J}_c - \hat{V}\|_{1,c} &\leq \|\Phi r' - \hat{V}\|_{1,c} \\ &\leq \|\Phi r' - \hat{V}\|_\infty \text{ (Since } c \text{ is a distribution)} \\ &\leq \|\Phi r' - J^*\|_\infty + \|J^* - \hat{V}\|_\infty \quad \text{(From Corollary 23 we have)} \\ &\leq \frac{4\gamma + \beta}{1 - \alpha} \end{aligned} \quad (32)$$

**Corollary 1** Let  $\hat{J}_c, \hat{V}, r^*$  and  $J^*$  be as in Theorem 25, then

$$\|J^* - \hat{J}_c\|_{1,c} \leq \frac{6\|J^* - \Phi r^*\|_\infty + 2\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty}{1 - \alpha}. \quad (33)$$

**Proof:**

$$\begin{aligned} \|J^* - \hat{J}_c\|_{1,c} &\leq \|J^* - \hat{V}\|_{1,c} + \|\hat{V} - \hat{J}_c\|_{1,c} \\ &\leq \|J^* - \hat{V}\|_\infty + \|\hat{V} - \hat{J}_c\|_{1,c} \end{aligned}$$

The result is obtained by using Corollary 23 for the first term and Theorem 25 for the second term in the above inequality.

## B Numerical Example of Single Queue with Finite Buffer size and Controlled Service Rates

The problem setting we consider is similar to the one presented in Sections 5.2 and 6.1 in [6]. However, we provide the most important details in this section so as to make the material self contained.

We consider a single queue with finite buffer size where the maximum allowed queue length is  $n - 1$ . The queue evolves in discrete instants of time  $t = 0, 1, \dots$  with only one of the following mutually exclusive events occurring between  $t$  and  $t + 1$

- A job arrives with probability  $p$ .
- A job gets served and leaves the queue with probability  $q(a)$ . Here  $a \in A = \{1, \dots, d\}$  is an action.

It is understood that excess jobs (i.e., jobs arriving when the queue length is  $n - 1$ ) will be discarded.

Formally, the dynamics of the controlled queuing system can be described via the framework of Markov Decision Process (MDP). The state space is given by  $S = \{0, 1, \dots, n - 1\}$  and denotes the number of jobs waiting in the queue. The action set is given by  $A = \{1, \dots, d\}$  and controls the probability of a job getting serviced and leaving the queue. We let  $s_t$  denote the state at time  $t$ . At time  $t$ , for  $0 < s_t < n - 1$ , the state at time  $t + 1$  when action  $a_t \in A$  is chosen is given by

$$\begin{aligned} s_{t+1} &= s_t + 1, \text{ with probability } p, \\ &= s_t - 1, \text{ with probability } q(a_t), \\ &= s_t, \text{ with probability } (1 - p - q(a_t)). \end{aligned} \quad (34)$$

For states  $s_t = 0$  and  $s_t = n - 1$  the system dynamics is given by

$$\begin{aligned} s_{t+1} &= s_t + 1, \text{ with probability } p, \text{ when } s_t = 0 \\ &= s_t - 1, \text{ with probability } q_{a_t}, \text{ when } s_t = n - 1. \end{aligned} \quad (35)$$

In order to ensure ‘stabilizability’, we assume the following condition on the system:

$$0 < q(1) \leq \dots \leq q(d) < 1, \text{ where } q(d) > p. \quad (36)$$

Note that the above state transition description in (34) and (35) has been presented in a concise format in Section 5. The reward associated with the action  $a \in A$  in state  $s \in S$  is given by

$$g_a(s) = -(s + 60q(a)^3). \quad (37)$$

The reward function is negative in queue length since it is desirable to penalize higher queue length. One can also observe that (37) penalizes actions that offer higher level of service.

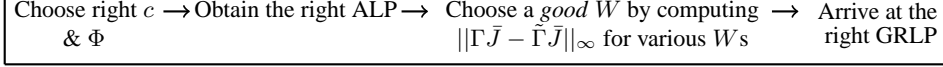


Figure 4: A step by step method to arrive at the right GRLP.

## C Solution via GRLP

We present the solution methodology in Figure 4.

### C.1 Choice of $\Phi$ and $c$

The polynomial features are known to work well for this problem. [6] gives a proper justification of this choice using arguments based on Lyapunov function. A good choice of  $c$  is

$$c(s) = (1 - \zeta)\zeta^s, \forall s = 0, 1, \dots, n - 1. \quad (38)$$

$c$  is the state relevance weight vector and denotes the relative importance of the various states. By choosing  $c$  as in (38), one can give importance to smaller queue lengths compared to large queue lengths. This choice is also supported by the fact that the stationary probability  $\pi(s)$  of the state  $s$  in the case of a stable uncontrolled queue is of the form  $\pi(s) \propto (\frac{\rho}{1-\rho})^s$ . However, when  $n$  is small (say 10)  $c$  can have a uniform distribution, since the ratio of  $\frac{c(n)}{c(0)} = \zeta^n$  is close to 1 for  $\zeta$  sufficiently close to 1 (say  $\zeta = 0.99$ ), i.e., the state relevance weights do not decay much for small  $n$ .

### C.2 Choice of $W$

Grouping adjacent states as in (14) (presented in the main body of the paper) might be a good idea. This choice of  $W$  was validated by Table 1 and we provide further insights by presenting the active and passive constraints of both ALP and GRLP for smaller system  $Q_S$  in Figure 5.

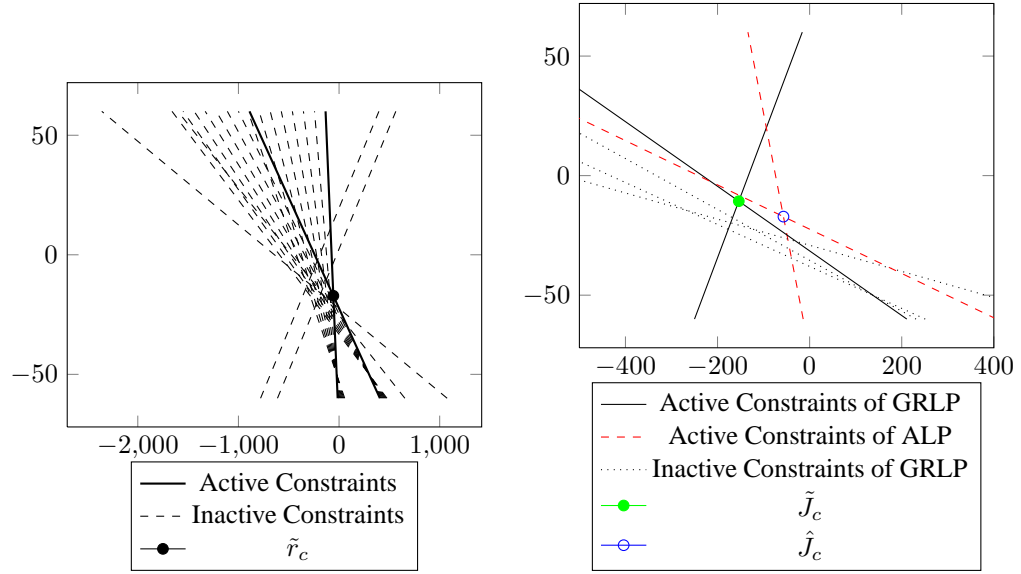
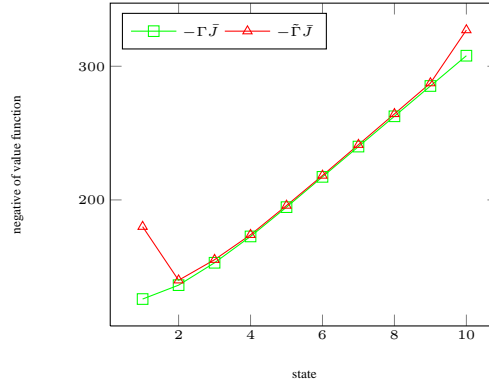
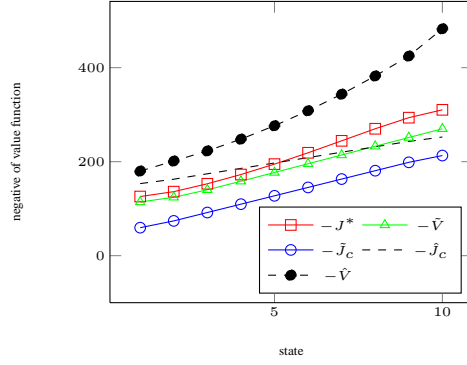


Figure 5: Constraints of the ALP (left) and the GRLP for system  $Q_S$  with  $n = 10$ ,  $d = 2$ ,  $k = 2$ ,  $m = 5$ ,  $q(1) = 0.2$ ,  $q(2) = 0.4$ ,  $p = 0.2$  and  $\alpha = 0.98$ . In this case  $c$  has a uniform distribution. The dotted and solid lines in the right plot show the inactive and active constraints of the GRLP respectively, the dashed lines in the right plot show the active constraints of the ALP. The feasible region in both cases (ALP & GRLP) are to the right of the corresponding active constraints.

The following plot shows the functions related to  $\|\Gamma \bar{J} - \tilde{\Gamma} \bar{J}\|_\infty$  (for system  $Q_S$ ):



The following plot shows the various error terms (for system  $Q_S$ ):



### C.3 Performance of Greedy Policy

We define the greedy policy  $\hat{u}$  as the one which is greedy with respect to  $\hat{J}_c$ , i.e.,

$$\hat{u}(s) = \arg \max_{a \in A} (g_a(s) + \alpha \sum_{s'} p_a(s, s') \hat{J}_c(s')). \quad (39)$$

The following plot shows the performance of  $\hat{u}$  in the case of the two systems ( $Q_S$  and  $Q_L$ ).

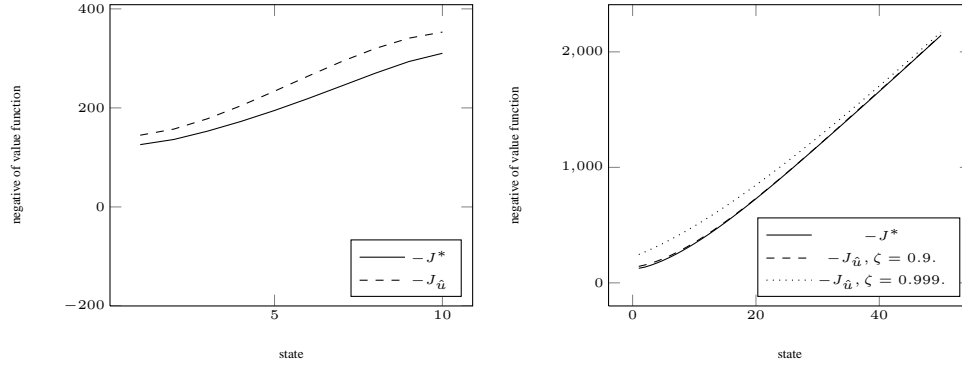


Figure 6: Plot corresponding to  $Q_S$  on the left and  $Q_L$  on the right.

It can be seen from the figures that  $J_{\hat{u}}$  is close to  $J^*$ . In particular, in the case of  $Q_L$ ,  $J_{\hat{u}}$  is nearly the same as  $J^*$  for  $\zeta = 0.999$ .